Progress Report Two

Arana Charlebois, Tristan
Carvalho, Cam
Corbett, Cole
Kassie, Turner
Pauls, Andrew
Wallace, Jordan

Cosc 4P02
Dr. Naser Ezzati-Jivan

March 23, 2025

Contents

1 Introduction

Before presenting the progression of the Logistics System, we would like to provide a brief reminder of the goal and benefits of such a system. The goal is to compute a viable travel route for a user, based on locations, number of drivers (for companies) and additional route options such as avoiding highways. As mentioned, such a system is of value to logistics and courier companies, since it offers a way for a company to organize a team of delivery drivers. Furthermore, the system can assist users who need an optimized route to complete daily tasks efficiently. What we plan to complete for the 4P02 project is truly just the beginning of a system of this kind. The idea has high potential in numerous areas of logistics.

Within our previous submissions for the 4P02 project, we outlined the purpose of our Logistics System, requirements/goals for the system as well as providing an update in regards to the development of our system in the case of progress report one. The purpose of this document is to provide another update on the current state of our Logistics System, particularly focusing on the improvements to the UI and addition of multi-TSP capabilities. Along with information about the system's capabilities, we will also outline major challenges during development, changes in development caused by an adjustment of requirements, group member contributions to the overall project and the Github activity for the groups operational repositories.

2 System Details and System State

Like in the previous report, the following section will be presented in two parts, the frontend progression and the backend progression. However, in contrast to the first report, we would like to showcase the backend, then the frontend, as the advancements on the backend will segment well into the frontend's growth.

2.1 Backend Development

In this subsection, we want to clealy explain the functionality of the backend, as there is no real user interface for this subsystem. Previously, we had completed and presented the content in regards to a GA for solving single-TSP. Shortly after, we began the task of adapting our existing solution for single-TSP to be a viable one for multi-TSP. Our original plan was to implement K-Medoids clustering using javaML to partition nodes, however, this lead to inconsistency in the results of the library and was ultimately removed. Naturally, this wasn't acceptable, and from the work completed by Cole, we were able to make use of the Google Or-tools and a distance matrix computed by the Graphhopper library to partition nodes and find an ordering. The Google Or-tools library has essentially provided a solution for the Multi-TSP aspect of the backend of the system, and along with adjustments to

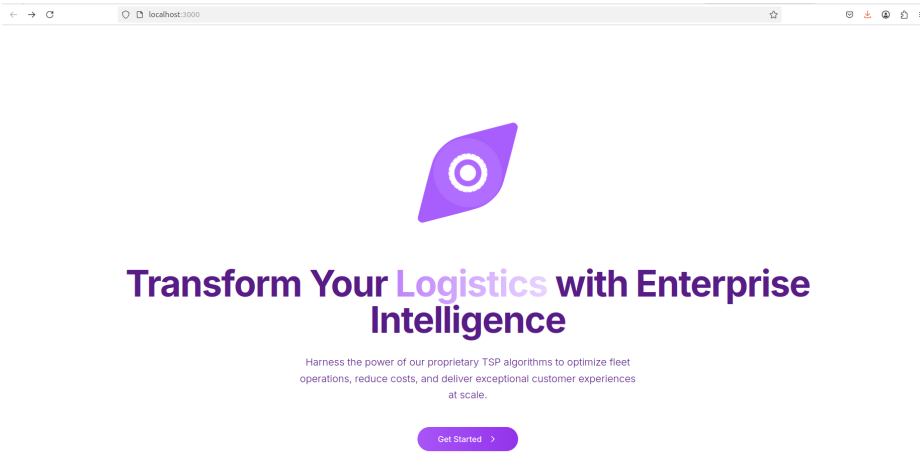
the already existing GA, we consider this aspect of the system complete.

The next area of the backend of the system that we want to highlight is the addition of flags for a given route, making use of the capabilities of Graphhopper. We have added the ability to read in these flags from an input file along with the locations to generate a route. Flags include options such as avoiding highways, tolls, unpaved roads, ferries, and train tracks, allowing the users to control more regarding the final route. These options are reflected in the distance matrix when using multi-TSP and the fitness function when using single-TSP.

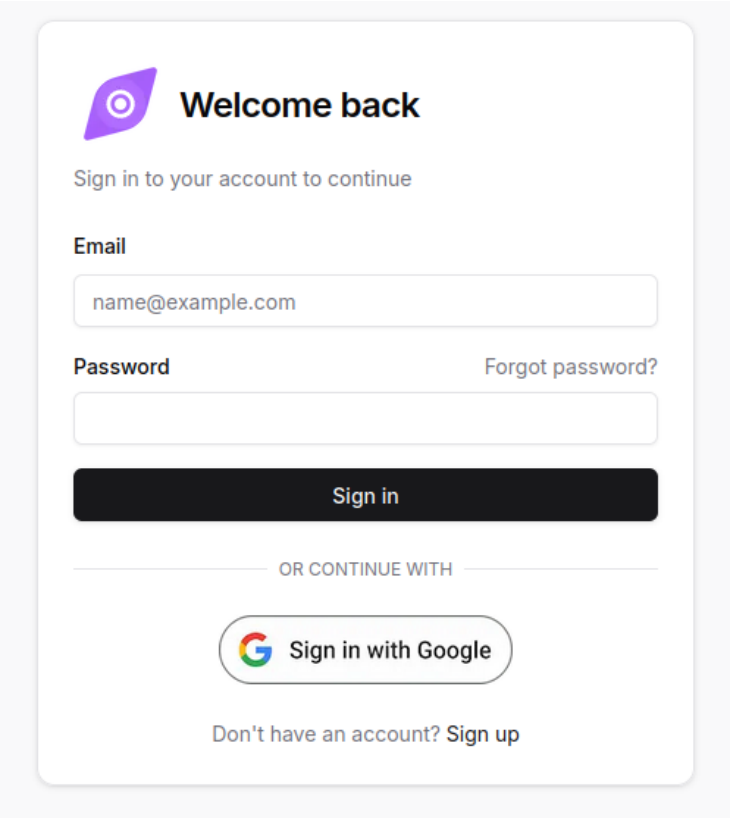
Finally, we show the update to the overall architecture of the backend of the system. Both inputs and outputs are in the form of GeoJson files. Inputs and outputs can account for the user seeking a single-TSP or multi-TSP solution (addition of driver-ID to a location), as well as ensuring that the output lists the proper stops for a vehicle route returning to the starting location or not. Also, when the code is deployed on a server, it waits for a request from a user, and passes this information to a thread to complete the task, ensuring that the server has no need to rebuild the libraries repeatedly saving time.

2.2 Frontend Development

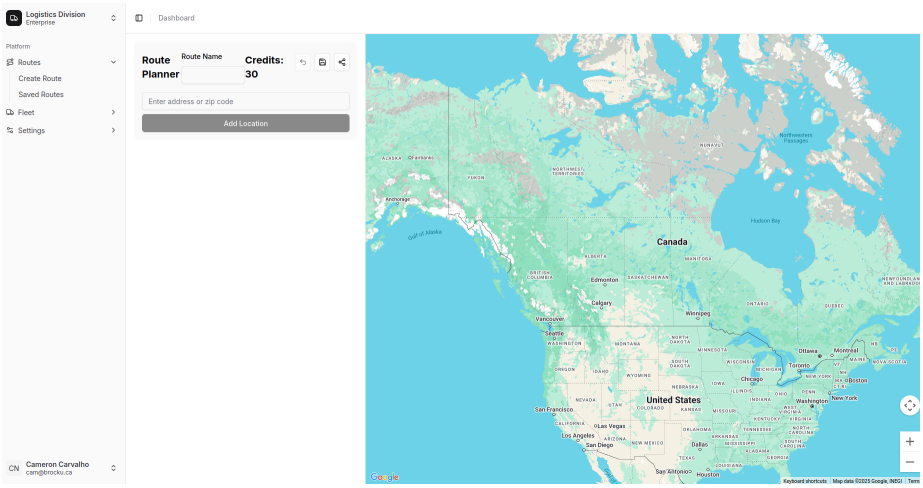
Development on the frontend is now the main focus of the group's time, as this has naturally alot of work to complete and needs to reflect the completion of multi-TSP on the backend. We show the current state of the frontend, starting with the landing page.



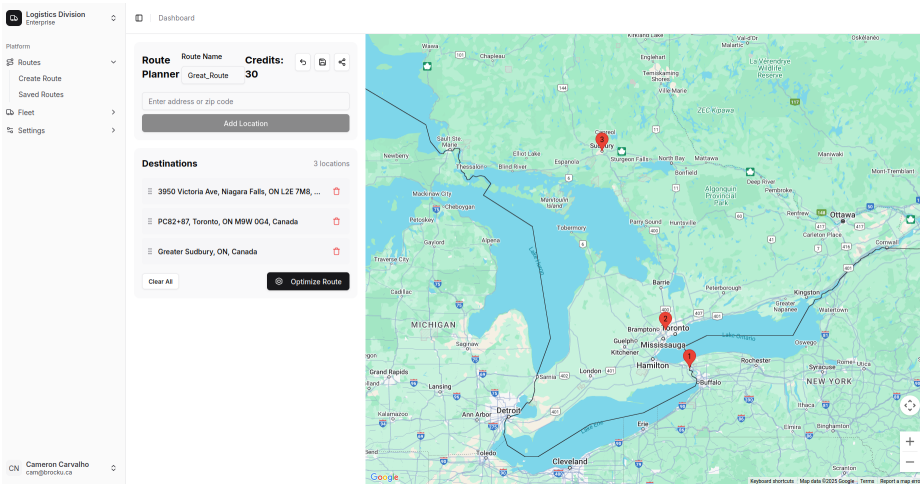
Just as before, the users can sign into their accounts via login page.

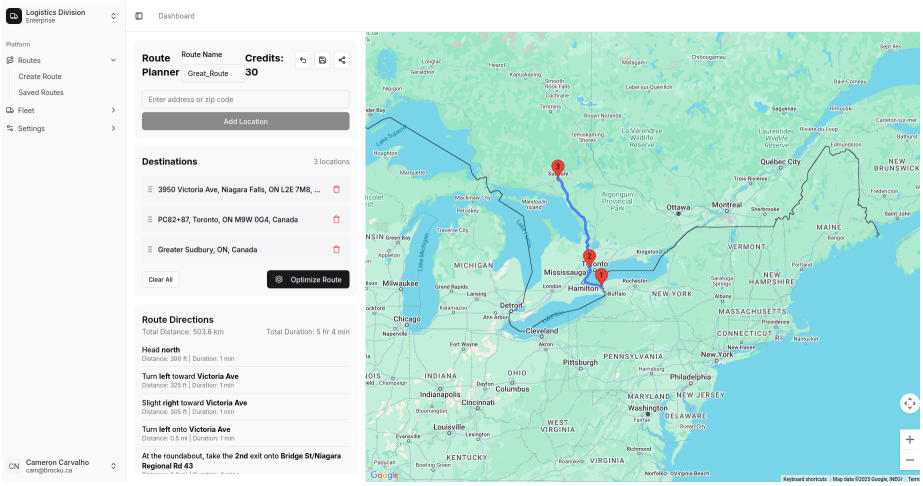


The dashboard has progressed to reflect changes made to the requirements in section 4, and other aspects of the system that had yet to be added in the previous report, such as a credit system.

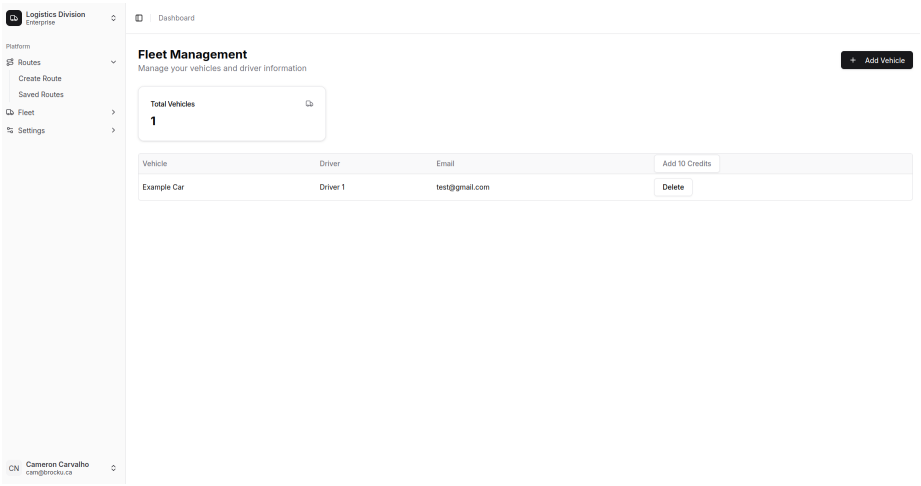


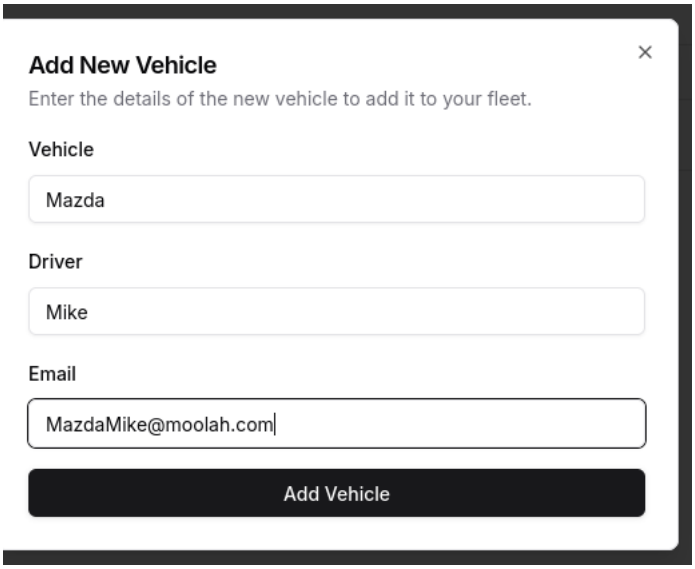
Next, we show the routing features of the frontend. Just as before, the ordering of locations is adjusted based on the route returned by the GA, but now, we note the addition of the directions associated with a route. This functionality can be seen in the two following screenshots;

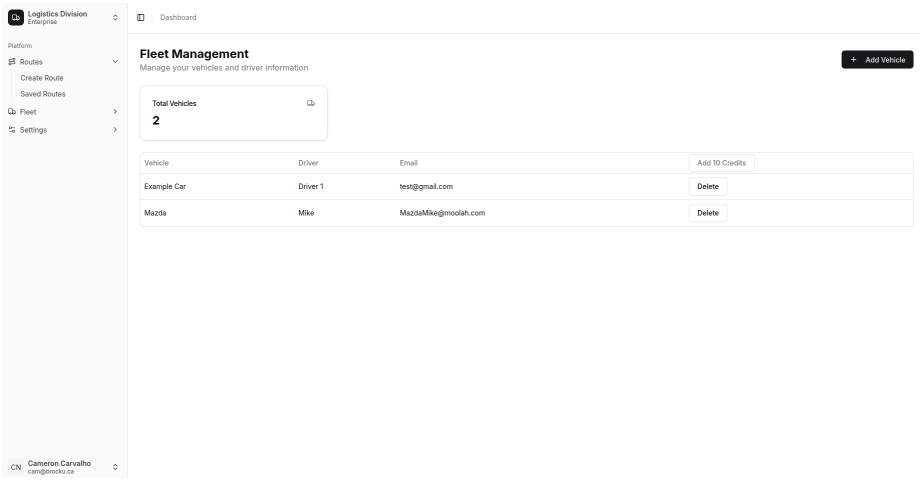




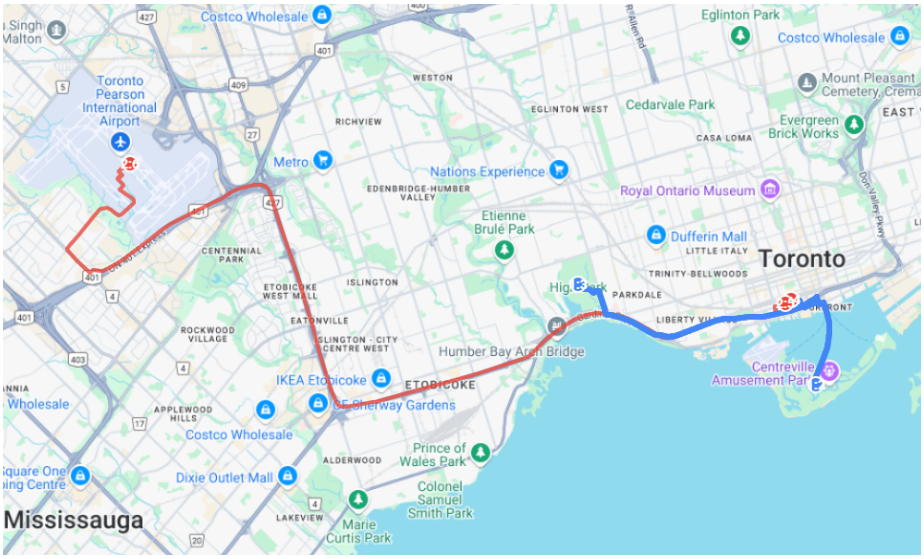
The fleet page was adjusted to be consistent with the altered requirements in section 4. The following shows the functionality of the fleet page, with the addition of a driver to a user's fleet.


Logistics Division
Enterprise
Platform
Routes
Create Route
Saved Routes
Fleet
Settings
CN
Cameron Carvalho
cam@brocku.ca
Dashboard
Fleet Management
Manage your vehicles and driver information
Add Vehicle
Total Vehicles
1
Vehicle
Driver
Email
Add 10 Credits
Example Car
Driver 1
test@gmail.com
Delete


Add New Vehicle
Enter the details of the new vehicle to add it to your fleet.
Vehicle
Mazda
Driver
Mike
Email
MazdaMike@moolah.com
Add Vehicle



Finally, while not complete, we have the preliminary steps completed for drawing lines for routes in a multi-TSP situation. While this is currently using hard coded data, it shows the main idea of how the system will behave for drawing multiple lines.



Other additions to the frontend include the addition of a logout button, multi-TSP directions, and the addition of flags that the user can set to be consistent with Graphhopper capabilities on the backend.

3 Challenges During Development

In this section, we give itemized lists of challenges faced while developing both subsystems. Naturally, the list will be broken into two lists, one for each subsystem.

3.1 Backend Challenges

- Finding tools that can integrate with other libraries, the current system, and are useful for completing the system.
- Adding standardized input and output, and integrating this with the libraries being used and ensuring that the output can be used by the frontend of the system.
- Continuous performance enhancements, for example, not building multiple routes for different conditions in the Graphhopper instance, and only building the one for the conditions that the user selects or keeping the server warm. (Major improvements found by Cole).

3.2 Frontend Challenges

- Keeping minimal branches, as many group members are constantly pushing and making changes as the system progresses.
- Utilizing API calls properly, and being efficient about when and how they're being used as this has real world costs.
- Properly structuring data for saving routes within the DB.
- Writing code for handling the backend output as the middle ware is still being installed.

4 Changes in Requirements

As mentioned earlier in this report, changes were made to the system requirements. The reason for the changes was due to a choice of aiming to complete a system that is polished, and shows consistent functionality throughout the system, rather than one that contains incomplete features. A group of team members (Andrew, Cam, Tristan and Turner) took time to meet (on March 4th, 2025) to redefine the functions of the system and provide clear and concise requirements that can be completed. The requirement changes were recorded and are listed below, in two sections, namely requirement redefinitions and requirements cuts. Note that the bullets listed are the major points of the meeting, many elements of the system were discussed, and small, minor changes were also planned, however, the aspects of the system relating to the minor changes isn't the focus of the report. The

submission of the document specifying the altered requirements and meeting itself will be submitted with the final submission of the work.

4.1 Requirement Redefinitions

- Each user is a standalone entity, and when a user makes an account, company information is tied to it if needed. They can save a driver name and email if needed too.
- Once routes can be isolated, a link for that route can be exported (rather then viewing within the application as another user).
- Landing page to be remade to be consistent with the overall system.
- Addition of billing page, so that user credit system can be navigated easily.

4.2 Cut Requirements

- Logistics division and enterprise functionality. As noted above, a user now only keeps a list of contacts, and can export routes to them.
- User having the option to add/remove drivers when the route has already been computed (and also any other adjustments to the route).
- Notifications.
- Settings page and all sub-pages are to be cut in place of a billing page. The settings page contained mostly information regarding enterprises, which has cut functionality too.

5 Conclusions and Next Steps

Now, we discuss the next plans in the development of the Logistics System and wrapping up the project. Between the previous progress report and this one, overall, we believe that we have made good progression. Multi-TSP has been implemented on the backend, message passing between the front and backend of the system is standardized (waiting on the middleware), and the frontend has seem numerous improvements as seen in section 2. A main factor in the development of the system was that collectively, as a group we agreed that having a complete system that has been tested is an overall goal, and in turn, we needed to reduce the scope of the overall project. Our focus is on two main aspects; completing the polyline logic for multi-TSP as well as setting up the middleware for message passing between the front and backend.

After the logic for polylines and middleware, we want to take time before testing and system validation to ensure that the product that we have is in line with our current, adjusted requirements. This will then lead into system

testing and validation, before the presentation and report. Overall, while for about two weeks after the submission of the first report, development continued as planned, we needed to reevaluate the situation that we were in, and take time to work on the requirements, and more time afterwards to develop so that the system will be in a form that is viable to launch. As a group, we are happy with the work that has been completed and the direction that the project in heading after feeling slightly unsure about the state of the system.

6 Group Member Contributions

Below is the work completed by each respective member (Note that the description of the work completed by each member is given by the member themselves);

6.1 Arana Charlebois, Tristan

- Implemented a MongoDB database hosted on Atlas to store user account information, and eventually routes generated by our GA.
- Registration and login flow prototype using Next.js was developed, implementing NextAuth.js as middleware to secure routes and store user data safely.
- Third-party OAuth providers like Google were implemented to allow users to login with already existing accounts, and a password reset flow via nodemailer was made to allow users to reset their password by following a link sent to their email.
- Added the ability to name and save routes into the MongoDB database, and load routes back to the frontend to display on the Google Maps view.
- Created fleet page to store information about vehicles, drivers, and their relevant contact information.
- Implemented credit system where users can add credits to their account. Performing actions such as saving and loading routes will deplete their credits based on the complexity of the route.

6.2 Carvalho, Cam

- Set up a VPS to handle our deployments and built a route calculation system that can scale with user growth using RabbitMQ.
- Created initial project backlog and initial sprints on Trello.

- Ported local systems to a dedicated server and configured services like MongoDB, Apache Kafka. Also connected the backend of the system to the frontend for single-TSP and currently completing for multi-TSP.
- Added stripe billing support for in-web purchasing of user credits.
- Deployed new version of the microservice for route to work with the new multi-TSP GA.

6.3 Corbett, Cole

- Researched and identified required libraries and data for our system and created the corresponding Gradle dependencies file as well as a template so other group members can use the identified libraries.
- Refactored the GA for readability and to better align with a standard GA implementation.
- Added elitism to the GA.
- Added graphing for visualization of the overall fitness of the population and testing purposes to the GA.
- Implemented K-Medoids clustering using javaML before switching completing the implementation of multi-TSP using google Or-Tools.
- Created unified program entry point for Cam and Andrew to easily connect the messenger service (RabbitMQ).
- Added support for all 32 additional profile configurations originating from the route options.
- Restructuring of prebuilt libraries, and restructure of code to keep the server warm waiting for a request, improving overall efficiency (sped up the API by 20x in when testing select inputs).

6.4 Kassie, Turner

- Authored both initial draft and final submission of all documents required in the course (project proposal, release planning document, progress report one and progress report two).
- Developed the initial GA used for solving single-TSP, that is the implementation of all core methods in a standard GA.
- Added reading from and writing to standard GeoJson file format allowing the GA to pass information to and from other components within the system.

- Restructured GA classes and methods to handle new route conditions (such as a fitness function that accounts for returning to the starting location).
- Restructured GA input and output classes and methods to account for newly added route flags as well as account for the multi-TSP libraries, added by Cole.

6.5 Pauls, Andrew

- Created initial project backlog and initial sprints on Trello.
- Maintained meeting minutes, recorded the adjusted MVP specifications in adaptation to a tighter timeline than anticipated.
- Amalgamated three branches into one simplified branch which included a combination Tristan's, Jordan's, and Andrew's frontend changes.
- Added changes to the overall UI, including the addition of logout button on dashboard returning the user to the landing page, a simplified landing page and links between the landing page and to authorization pages (in both directions).
- Wrote functions that can read, parse, and handle multi-TSP JSON data from the format specified by Turner.

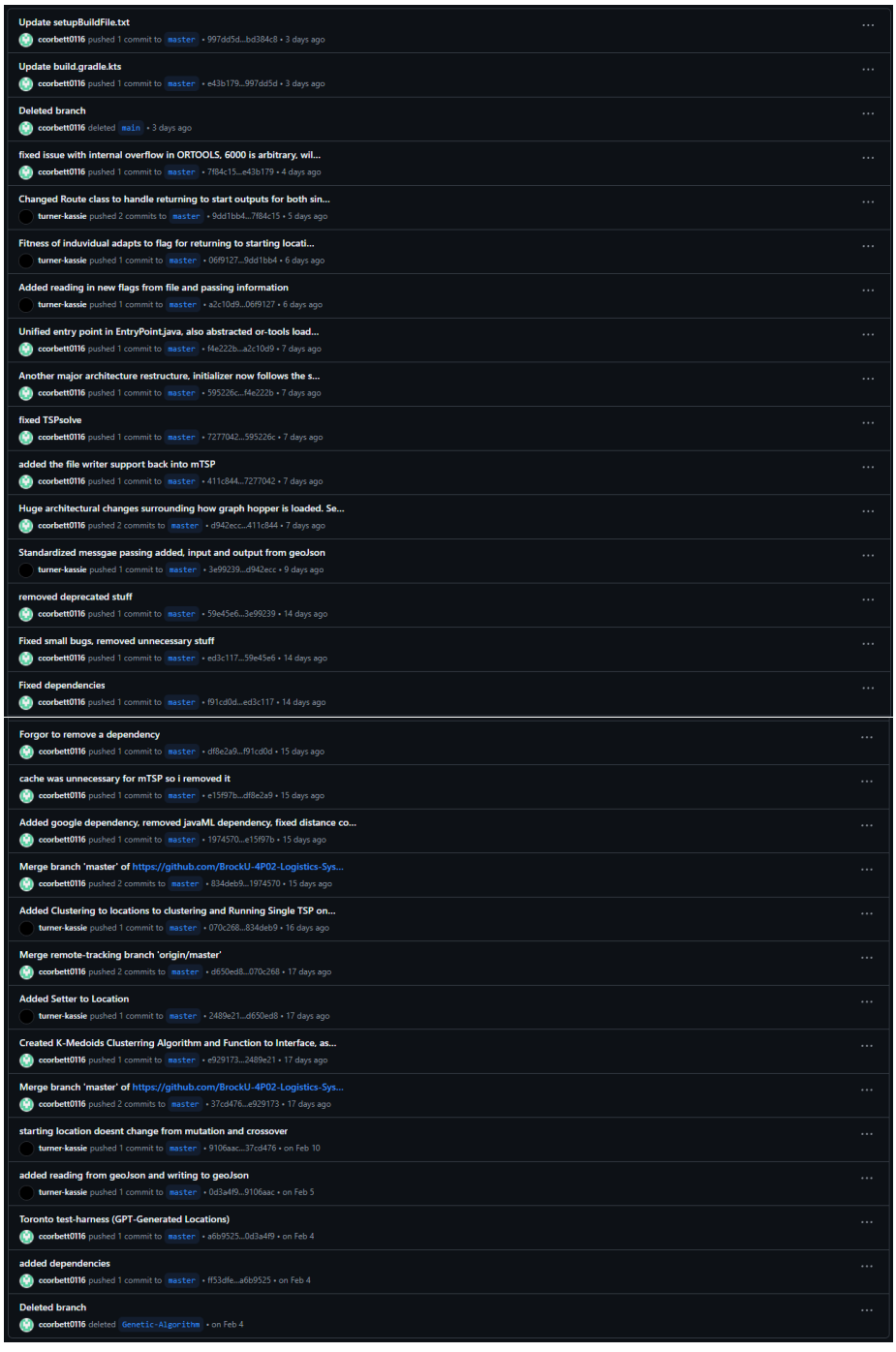
6.6 Wallace, Jordan

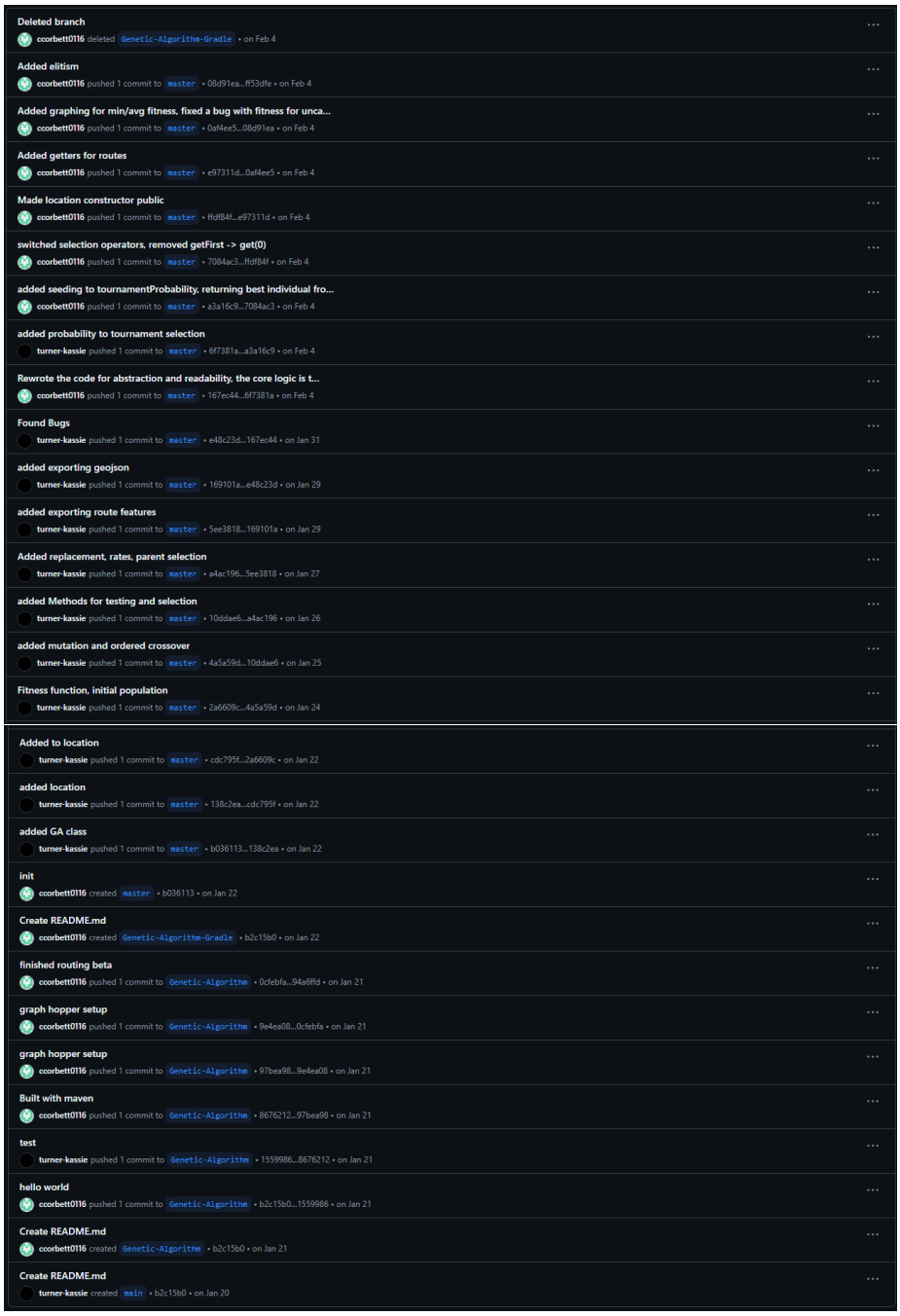
- Created the logo for the product in Illustrator and Photoshop, later completing a revision of said logo.
- Utilized Google Maps API to allow users to pick auto-completed addresses within Canada when searching a particular location in app.
- Assigned a marker to to given addresses/locations so the user has a visual representation of their route.
- Added directions panel for a selected route, and later updated to account for multi-TSP and users selecting the respective routes.
- Implemented frontend component of the new multi-TSP code which draws an individual poly line for each route.
- Added ability to assign desired number of drivers/vehicles to dashboard.

7 Github Logs

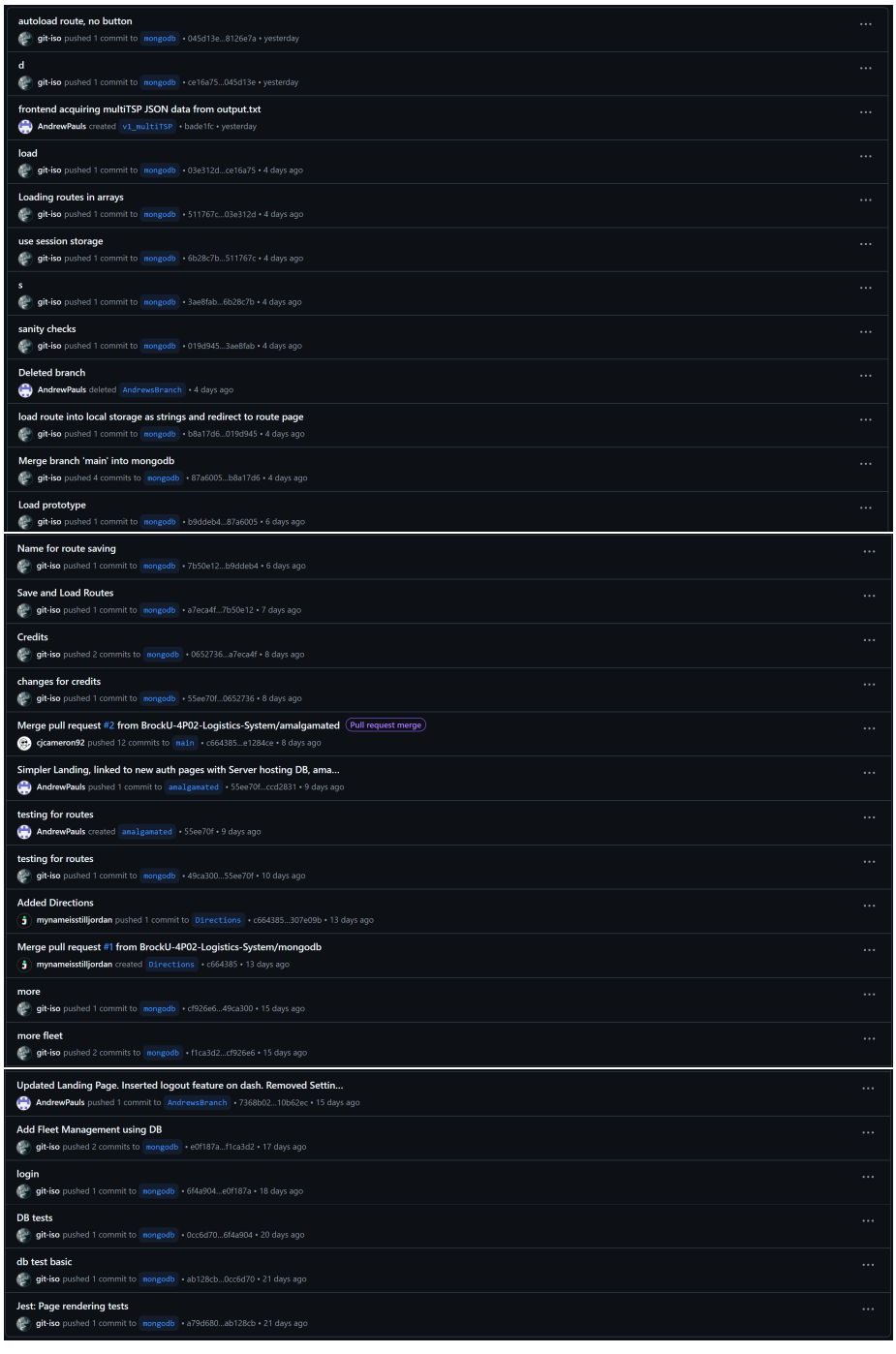
The Github logs for each repository within our Logistics System organization are within their respective subsections. The screenshots recorded were taken on March 22nd, 2025 for the readers reference. Also, note that the repositories included in this section are the ones currently being utilized by the group.

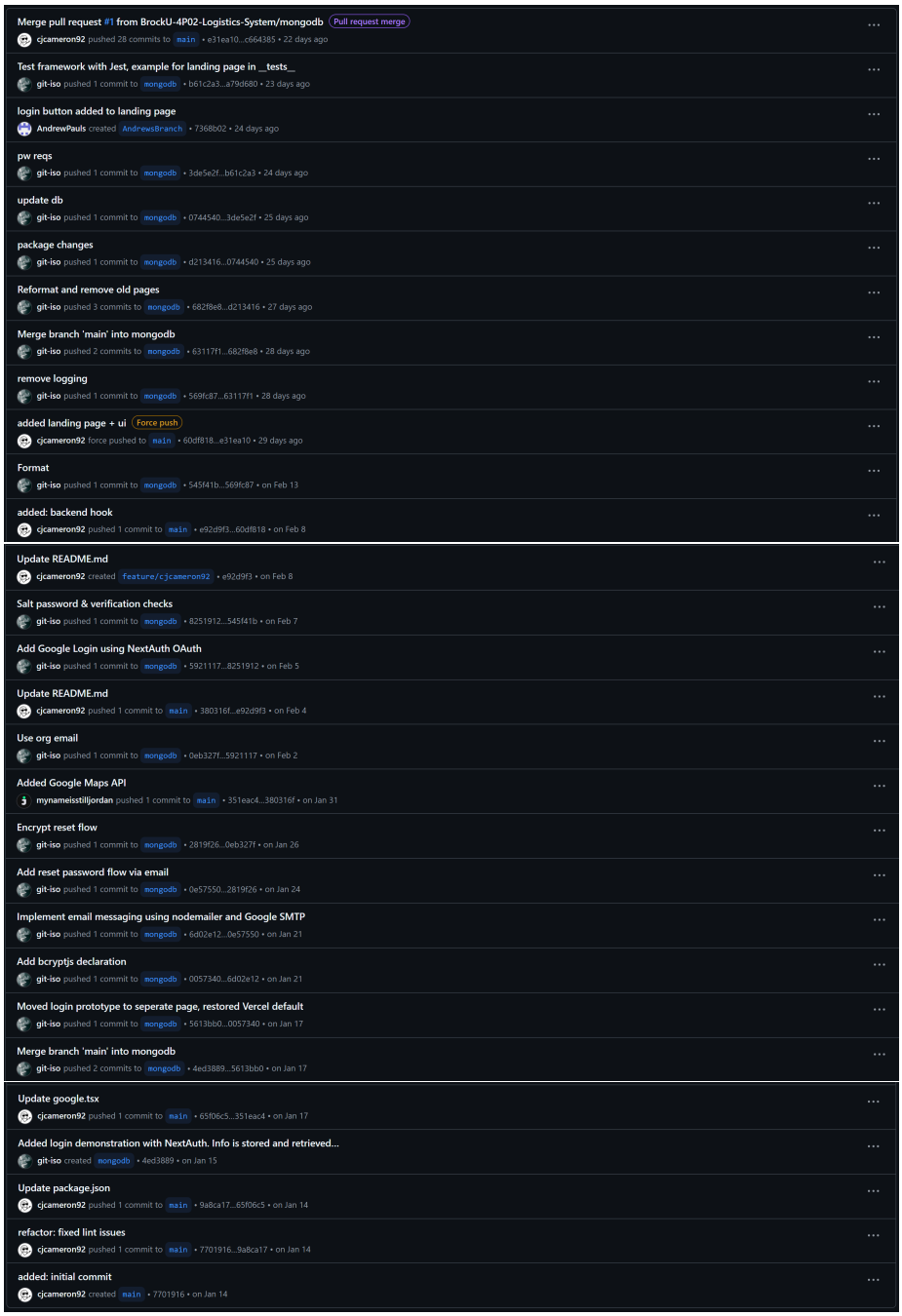
7.1 Logistics System Repository




Deleted branch
ccorbett0116 deleted Genetic-Algorithm-Gradle • on Feb 4
Added elitism
ccorbett0116 pushed 1 commit to master • 08d91ea...ff53dfe • on Feb 4
Added graphing for min/avg fitness, fixed a bug with fitness for unca...
ccorbett0116 pushed 1 commit to master • 0af4ee5...08d91ea • on Feb 4
Added getters for routes
ccorbett0116 pushed 1 commit to master • e97311d...0af4ee5 • on Feb 4
Made location constructor public
ccorbett0116 pushed 1 commit to master • ffdf84f...e97311d • on Feb 4
switched selection operators, removed getFirst -> get(0)
ccorbett0116 pushed 1 commit to master • 7084ac3...ffdf84f • on Feb 4
added seeding to tournamentProbability, returning best individual fro...
ccorbett0116 pushed 1 commit to master • a3a16c9...7084ac3 • on Feb 4
added probability to tournament selection
turner-kassie pushed 1 commit to master • 6f7381a...a3a16c9 • on Feb 4
Rewrote the code for abstraction and readability, the core logic is t...
ccorbett0116 pushed 1 commit to master • 167ec44...6f7381a • on Feb 4
Found Bugs
turner-kassie pushed 1 commit to master • e48c23d...167ec44 • on Jan 31
added exporting geojson
turner-kassie pushed 1 commit to master • 169101a...e48c23d • on Jan 29
added exporting route features
turner-kassie pushed 1 commit to master • 5ee3818...169101a • on Jan 29
Added replacement, rates, parent selection
turner-kassie pushed 1 commit to master • a4ac196...5ee3818 • on Jan 27
added Methods for testing and selection
turner-kassie pushed 1 commit to master • 10ddae6...a4ac196 • on Jan 26
added mutation and ordered crossover
turner-kassie pushed 1 commit to master • 4a5a59d...10ddae6 • on Jan 25
Fitness function, initial population
turner-kassie pushed 1 commit to master • 2a6609c...4a5a59d • on Jan 24
Added to location
turner-kassie pushed 1 commit to master • cdc795f...2a6609c • on Jan 22
added location
turner-kassie pushed 1 commit to master • 138c2ea...cdc795f • on Jan 22
added GA class
turner-kassie pushed 1 commit to master • b036113...138c2ea • on Jan 22
init
ccorbett0116 created master • b036113 • on Jan 22
Create README.md
ccorbett0116 created Genetic-Algorithm-Gradle • b2c15b0 • on Jan 22
finished routing beta
ccorbett0116 pushed 1 commit to Genetic-Algorithm • 0cfebfa...94a6ffd • on Jan 21
graph hopper setup
ccorbett0116 pushed 1 commit to Genetic-Algorithm • 9e4ea08...0cfebfa • on Jan 21
graph hopper setup
ccorbett0116 pushed 1 commit to Genetic-Algorithm • 97bea98...9e4ea08 • on Jan 21
Built with maven
ccorbett0116 pushed 1 commit to Genetic-Algorithm • 8676212...97bea98 • on Jan 21
test
turner-kassie pushed 1 commit to Genetic-Algorithm • 1559986...8676212 • on Jan 21
hello world
ccorbett0116 pushed 1 commit to Genetic-Algorithm • b2c15b0...1559986 • on Jan 21
Create README.md
ccorbett0116 created Genetic-Algorithm • b2c15b0 • on Jan 21
Create README.md
turner-kassie created main • b2c15b0 • on Jan 20

7.2 Frontend Repository




Merge pull request #1 from BrockU-4P02-Logistics-System/mongodb
Pull request merge
cjcameron92 pushed 28 commits to main • e31ea10...c664385 • 22 days ago
Test framework with Jest, example for landing page in __tests__
git-iso pushed 1 commit to mongodb • b61c2a3...a79d680 • 23 days ago
login button added to landing page
AndrewPauls created AndrewsBranch • 7368b02 • 24 days ago
pw reqs
git-iso pushed 1 commit to mongodb • 3de5e2f...b61c2a3 • 24 days ago
update db
git-iso pushed 1 commit to mongodb • 0744540...3de5e2f • 25 days ago
package changes
git-iso pushed 1 commit to mongodb • d213416...0744540 • 25 days ago
Reformat and remove old pages
git-iso pushed 3 commits to mongodb • 682f6e8...d213416 • 27 days ago
Merge branch 'main' into mongodb
git-iso pushed 2 commits to mongodb • 63117f1...682f8e8 • 28 days ago
remove logging
git-iso pushed 1 commit to mongodb • 569fc87...63117f1 • 28 days ago
added landing page + ui
Force push
cjcameron92 force pushed to main • 60df818...e31ea10 • 29 days ago
Format
git-iso pushed 1 commit to mongodb • 545f41b...569fc87 • on Feb 13
added: backend hook
cjcameron92 pushed 1 commit to main • e92d9f3...60df818 • on Feb 8
Update README.md
cjcameron92 created feature/cjcameron92 • e92d9f3 • on Feb 8
Salt password & verification checks
git-iso pushed 1 commit to mongodb • 8251912...545f41b • on Feb 7
Add Google Login using NextAuth OAuth
git-iso pushed 1 commit to mongodb • 5921117...8251912 • on Feb 5
Update README.md
cjcameron92 pushed 1 commit to main • 380316f...e92d9f3 • on Feb 4
Use org email
git-iso pushed 1 commit to mongodb • 0eb327f...5921117 • on Feb 2
Added Google Maps API
mynameisstilljordan pushed 1 commit to main • 351eac4...380316f • on Jan 31
Encrypt reset flow
git-iso pushed 1 commit to mongodb • 2819f26...0eb327f • on Jan 26
Add reset password flow via email
git-iso pushed 1 commit to mongodb • 0e57550...2819f26 • on Jan 24
Implement email messaging using nodemailer and Google SMTP
git-iso pushed 1 commit to mongodb • 6d02e12...0e57550 • on Jan 21
Add bcryptjs declaration
git-iso pushed 1 commit to mongodb • 0057340...6d02e12 • on Jan 21
Moved login prototype to seperate page, restored Vercel default
git-iso pushed 1 commit to mongodb • 5613bb0...0057340 • on Jan 17
Merge branch 'main' into mongodb
git-iso pushed 2 commits to mongodb • 4ed3889...5613bb0 • on Jan 17
Update google.tsx
cjcameron92 pushed 1 commit to main • 65f06c5...351eac4 • on Jan 17
Added login demonstration with NextAuth. Info is stored and retrieved...
git-iso created mongodb • 4ed3889 • on Jan 15
Update package.json
cjcameron92 pushed 1 commit to main • 9a8ca17...65f06c5 • on Jan 14
refactor: fixed lint issues
cjcameron92 pushed 1 commit to main • 7701916...9a8ca17 • on Jan 14
added: initial commit
cjcameron92 created main • 7701916 • on Jan 14